UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/764,495	01/27/2004	Andrew Mark Nightingale	550-510	6258

23117 7590 09/11/2007

NIXON & VANDERHYE, PC
901 NORTH GLEBE ROAD, 11TH FLOOR
ARLINGTON, VA 22203

EXAMINER	
ALHIJA, SAIF A	

ART UNIT	PAPER NUMBER
2128	

MAIL DATE	DELIVERY MODE
09/11/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/764,495
Filing Date: January 27, 2004
Appellant(s): NIGHTINGALE, ANDREW MARK

John Lastova
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 23 July 2007 appealing from the Office action mailed 5 January 2007.

(1) Real Party in Interest

A statement identifying by name the real part in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings, which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellants statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief are correct.

(8) Evidence Relied Upon

i) Rajsuman et al. "Method and Apparatus for SOC Design Validation", U.S. Patent No. 6,678,645, hereafter referred to as Rajsuman

(9) Grounds of Rejection

The following grounds of rejection are applicable to the appealed claims:

(9.1) Claims 1-51 are rejected because the claimed invention is directed to non-statutory subject matter.

i) Regarding Claims 1-51, the Examiner asserts that the current state of the claim language is such that a reasonable interpretation of the claims would not result in any useful, concrete or tangible result. The Examiner asserts that the claims do not indicate if the methods or apparatus are tangible methods or apparatus utilizing hardware, instead of an arrangement of software lacking tangible embodiment. Further, following Applicants amendment claim 35 recites an intended use. The phrase "for use in" in claim 35 is an intended use and therefore the limitations following said phrase are not afforded patentable weight.

ii) Regarding Claims 35-51, the claim recites a computer program. It should be noted that code (i.e., a computer software program) does not do anything per se. Instead, it is the code stored on a computer that, *when executed*, instructs the computer to perform various functions. The following claim is a generic example of a proper computer program product claim;

A computer program product embodied on a computer-readable medium and comprising code that, when executed, causes a computer to perform the following:

Function A
Function B
Function C, etc...

All Claims dependent upon a rejected a rejected base claim are rejected by virtue of their dependency.

(9.2) Claims 1-51 are rejected under 35 U.S.C. 102(e) as being clearly anticipated by **Rajsuman et al. "Method and Apparatus for SOC Design Validation", U.S. Patent No. 6,678,645,** hereafter referred to as **Rajsuman.**

Regarding Claim 1:

Rajsuman discloses Apparatus for performing a sequence of verification tests to perform hardware and software co-verification on a system under verification, comprising:

a plurality of signal interface controllers operable to be coupled to said system under verification, each signal interface controller being operable to perform one or more test actions transferring at least one of one or more stimulus signals and one or more response signals between a corresponding portion of the system under verification and said signal interface controller during performance of said sequence of verification tests; **(Column 5, Lines 43-44. Figure 5)**

a debugger operable to control operation of a processing unit associated with the system under verification, the processing unit being operable to execute software routines; **(Column 9, Lines 16-24. Column 11, Lines 53-63. Figure 5. Claim 5)**

a debugger signal interface controller operable to interface with the debugger and to perform one or more test actions transferring at least one of one or more stimulus signals and one or more response signals between the debugger and said debugger signal interface controller during performance of said sequence of verification tests; **(Column 9, Lines 16-24. Column 11, Lines 53-63. Figure 5. Claim 5)**

and a test manager coupled to said plurality of signal interface controllers and the debugger signal interface controller and operable to transfer test controlling messages to said plurality of signal interface controllers and the debugger signal interface controller identifying the test actions to be performed; **(Column 5, Lines 32-34. Figure 5)**

the test manager being operable to control the operation of the processing unit via the debugger signal interface controller and the debugger in order to co-ordinate the execution of said software routines with the sequence of verification tests. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 2:

Rajsuman discloses Apparatus as claimed in claim 1, further comprising a memory in which the software routines are stored, the debugger signal interface controller being provided with an address indication identifying the addresses of the software routines within the memory, upon receipt of a test controlling message from the test manager, the debugger signal interface controller being operable to generate a corresponding test action with reference to the address indication. **(Column 11, Lines 53-63. Figure 5)**

Regarding Claim 3:

Rajsuman discloses Apparatus as claimed in claim 2, further comprising a status memory operable to store status data, the processing unit being operable to execute monitoring code to monitor the status data in order to identify any changes to the status data and to then execute at least one of said software routines as identified by the change in status data, the corresponding test action being arranged to cause updated status data identifying a corresponding one of said software routines to be stored in the status memory under the control of the debugger. **(Column 11, Lines 53-63. Figure 5)**

Regarding Claim 4:

Rajsuman discloses Apparatus as claimed in claim 3, wherein the debugger is operable to cause the processing unit to store the updated status data in the status memory, whereafter the processing unit reverts to executing the monitoring code. **(Column 11, Lines 53-63. Figure 5)**

Regarding Claim 5:

Rajsuman discloses Apparatus as claimed in claim 2, wherein the processing unit has a plurality of registers associated therewith, and the corresponding test action is arranged to cause data in a selected register of said plurality of registers to be updated under the control of the debugger, such that the processing unit will then execute a corresponding one of said software routines. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 6:

Rajsuman discloses Apparatus as claimed in claim 5, wherein the selected register is operable to store a program counter value and the corresponding test action is arranged to cause the program counter value to be updated in order to cause the processing unit to branch to the corresponding one of said software routines. **(Column 12, Lines 15-35)**

Regarding Claim 7:

Rajsuman discloses Apparatus as claimed in claim 1, wherein upon execution of one of said software routines, the processing unit is operable to update status data indicative of whether the software routine completed successfully, the debugger signal interface controller being operable to perform a predetemmined test action in order to cause a breakpoint to be set by the debugger which is triggered when the processing unit performs said update of the status data. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 8:

Rajsuman discloses Apparatus as claimed in claim 7, wherein the debugger is operable to issue a callback event to the debugger signal interface controller upon triggering of the breakpoint. **(Column 12, Lines 15-35)**

Regarding Claim 9:

Rajsuman discloses Apparatus as claimed in claim 1, wherein said stimulus and/or response signals are transferred between the debugger signal interface controller and the debugger using Application Programming Interface (API) calls. **(Column 12, Lines 8-14)**

Regarding Claim 10:

Rajsuman discloses Apparatus as claimed in claim 2, wherein at least one of the software routines is written into the memory via the debugger under the control of the debugger signal interface controller. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 11:

Rajsuman discloses Apparatus as claimed in claim 1, wherein multiple processing units are provided, and a corresponding multiple of debugger signal interface controllers are provided, each debugger signal interface controller communicating with the same debugger to cause their respective test actions to be performed. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 12:

Rajsuman discloses Apparatus as claimed in claim 1, wherein the timing of the execution of said software routines is coordinated with the sequence of verification tests. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 13:

Rajsuman discloses Apparatus as claimed in claim 1, wherein the system under verification comprises a plurality of components, each signal interface controller being associated with one of said components. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 14:

Rajsuman discloses Apparatus as claimed in claim 13, wherein the processing unit forms one of the components of the system under verification. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 15:

Rajsuman discloses Apparatus as claimed in claim 1, further comprising the processing unit, the processing unit being provided externally to the system under verification. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 16:

Rajsuman discloses Apparatus as claimed in claim 1, wherein the processing unit comprises a representation of a processor on which the software routines are intended to be executed. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 17:

Rajsuman discloses Apparatus as claimed in claim 1, wherein the system under verification comprises a hardware simulator responsive to said one or more stimulus signals to generate said one or more response signals simulating a response of a data processing apparatus to said one or more stimulus signals if applied to said data processing apparatus. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 18:

Rajsuman discloses A method of performing a sequence of verification tests to perform hardware and software co-verification on a system under verification, comprising the steps of:

performing in each of a plurality of signal interface controllers coupled to said system under verification one or more test actions transferring at least one of one or more stimulus signals and one or more response signals between a corresponding portion of the system under verification and said signal interface controller during performance of said sequence of verification tests; **(Column 5, Lines 43-44. Figure 5)**

controlling via a debugger execution of software routines by a processing unit associated with the system under verification; **(Column 9, Lines 16-24. Column 11, Lines 53-63. Figure 5. Claim 5)**

performing in a debugger signal interface controller coupled with the debugger one or more test actions transferring at least one of one or more stimulus signals and one or more response signals between the debugger and said debugger signal interface controller during performance of said sequence of verification tests; **(Column 9, Lines 16-24. Column 11, Lines 53-63. Figure 5. Claim 5)**.

and transferring test controlling messages from a test manager to said plurality of signal interface controllers and to the debugger signal interface controller, the test controlling messages identifying the test actions to be performed; **(Column 5, Lines 32-34. Figure 5)**

whereby the test manager controls the operation of the processing unit via the debugger signal interface controller and the debugger in order to co-ordinate the execution of said software routines with the sequence of verification tests. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 19:

Rajsuman discloses A method as claimed in claim 18, further comprising the steps of:

storing the software routines in a memory; **(Column 11, Lines 53-63. Figure 5)**

providing the debugger signal interface controller with an address indication identifying the addresses of the software routines within the memory;

and upon receipt of a test controlling message from the test manager, generating in the debugger signal interface controller a corresponding test action with reference to the address indication. **(Column 11, Lines 53-63. Figure 5)**

Regarding Claim 20:

Rajsuman discloses A method as claimed in claim 19, further comprising the steps of:

storing status data in a status memory; **(Column 11, Lines 53-63. Figure 5)**

executing on the processing unit monitoring code to monitor the status data in order to identify any changes to the status data and then executing at least one of said software routines as identified by the change in status data, the corresponding test action causing updated status data identifying a corresponding one of said software routines to be stored in the status memory under the control of the debugger. **(Column 11, Lines 53-63. Figure 5)**

Regarding Claim 21:

Rajsuman discloses A method as claimed in claim 20, further comprising the step of: causing the processing unit to store the updated status data in the status memory, whereafter the processing unit reverts to executing the monitoring code. **(Column 11, Lines 53-63. Figure 5)**

Regarding Claim 22:

Rajsuman discloses A method as claimed in claim 19, wherein the processing unit has a plurality of registers associated therewith, and the corresponding test action causes data in a selected register of said

plurality of registers to be updated under the control of the debugger, such that the processing unit will then execute a corresponding one of said software routines. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 23:

Rajsuman discloses A method as claimed in claim 22, wherein the selected register is operable to store a program counter value and the corresponding test action causes the program counter value to be updated in order to cause the processing unit to branch to the corresponding one of said software routines. **(Column 12, Lines 15-35)**

Regarding Claim 24:

Rajsuman discloses A method as claimed in claim 18, further comprising the steps of:

upon execution of one of said software routines, employing the processing unit to update status data indicative of whether the software routine completed successfully; **(Column 5, Lines 41-48. Figure 5)**

and causing the debugger signal interface controller to perform a predetermined test action in order to cause a breakpoint to be set by the debugger which is triggered when the processing unit performs said update of the status data. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 25:

Rajsuman discloses A method as claimed in claim 24, wherein the debugger issues a callback event to the debugger signal interface controller upon triggering of the breakpoint. **(Column 12, Lines 15-35)**

Regarding Claim 26:

Rajsuman discloses A method as claimed in claim 18, wherein said stimulus and/or response signals are transferred between the debugger signal interface controller and the debugger using Application Programming Interface (API) calls. **(Column 12, Lines 8-14)**

Regarding Claim 27:

Rajsuman discloses A method as claimed in claim 19, further comprising the step of: writing at least one of the software routines into the memory via the debugger under the control of the debugger signal interface controller. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 28:

Rajsuman discloses A method as claimed in claim 18, wherein multiple processing units are provided, and a corresponding multiple of debugger signal interface controllers are provided, each debugger signal interface controller communicating with the same debugger to cause their respective test actions to be performed. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 29:

Rajsuman discloses A method as claimed in claim 18, wherein the timing of the execution of said software routines is coordinated with the sequence of verification tests. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 30:

Rajsuman discloses A method as claimed in claim 18, wherein the system under verification comprises a plurality of components, each signal interface controller being associated with one of said components. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 31:

Rajsuman discloses. A method as claimed in claim 30, wherein the processing unit forms one of the components of the system under verification. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 32:

Rajsuman discloses A method as claimed in claim 18, wherein the processing unit is provided externally to the system under verification. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 33:

Rajsuman discloses A method as claimed in claim 18, wherein the processing unit comprises a representation of a processor on which the software routines are intended to be executed. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 34:

Rajsuman discloses A method as claimed in claim 18, wherein the system under verification comprises a hardware simulator which in response to said one or more stimulus signals generates said one or more response signals simulating a response of a data processing apparatus to said one or more stimulus signals if applied to said data processing apparatus. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 35:

Rajsuman discloses A computer program product embodied on a computer-readable medium for use in performing a sequence of verification tests to perform hardware and software co-verification on a system under verification, the computer program product comprising:

a plurality of signal interface controller code blocks operable to be coupled to said system under verification, each signal interface controller code block being when executed causes a computer to perform one or more test actions transferring at least one of one or more stimulus signals and one or more response signals between a corresponding portion of the system under verification and said signal interface controller code block during performance of said sequence of verification tests **(Column 5, Lines 43-44. Figure 5)** ;

debugger code when executed causes the computer to control operation of a processing unit associated with the system under verification, the processing unit being operable to execute software routines; **(Column 9, Lines 16-24. Column 11, Lines 53-63. Figure 5. Claim 5)**

a debugger signal interface controller code block when executed causes the computer to interface with the debugger code and to perform one or more test actions transferring at least one of one or more stimulus signals and one or more response signals between the debugger code and said debugger signal interface controller code block during performance of said sequence of verification tests; **(Column 9, Lines 16-24. Column 11, Lines 53-63. Figure 5. Claim 5)**

and test manager code coupled to said plurality of signal interface controller code blocks and the debugger signal interface controller code block and when executed causes the computer to transfer test controlling messages to said plurality of signal interface controller code blocks and the debugger signal interface controller code block identifying the test actions to be performed; **(Column 5, Lines 32-34. Figure 5)**

the test manager code when executed causes the computer to control the operation of the processing unit via the debugger signal interface controller code block and the debugger code in order to co-ordinate the execution of said software routines with the sequence of verification tests. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 36:

Rajsuman discloses A computer program product as claimed in claim 35, wherein the software routines are stored in a memory, the debugger signal interface controller code block being provided with an address indication identifying the addresses of the software routines within the memory, upon receipt of a test controlling message from the test manager code, the debugger signal interface controller code block when executed causes the computer to generate a corresponding test action with reference to the address indication. **(Column 11, Lines 53-63. Figure 5)**

Regarding Claim 37:

Rajsuman discloses A computer program product as claimed in claim 36, wherein status data is stored in a status memory, the processing unit being operable to execute monitoring code to monitor the status data in order to identify any changes to the status data and to then execute at least one of said software routines as identified by the change in status data, the corresponding test action being arranged to cause updated status data identifying a corresponding one of said software routines to be stored in the status memory under the control of the debugger code. **(Column 11, Lines 53-63. Figure 5)**

Regarding Claim 38:

Rajsuman discloses A computer program product as claimed in claim 37, wherein the debugger code is operable to cause the processing unit to store the updated status data in the status memory, whereafter the processing unit reverts to executing the monitoring code. **(Column 11, Lines 53-63. Figure 5)**

Regarding Claim 39:

Rajsuman discloses A computer program product as claimed in claim 36, wherein the processing unit has a plurality of registers associated therewith, and the corresponding test action is arranged to cause data in a selected register of said plurality of registers to be updated under the control of the debugger code, such that the processing unit will then execute a corresponding one of said software routines. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 40:

Rajsuman discloses A computer program product as claimed in claim 39, wherein the selected register is operable to store a program counter value and the corresponding test action is arranged to cause the program counter value to be updated in order to cause the processing unit to branch to the corresponding one of said software routines. **(Column 12, Lines 15-35)**

Regarding Claim 41:

Rajsuman discloses A computer program product as claimed in claim 35, wherein upon execution of one of said software routines, the processing unit is operable to update status data indicative of whether the software routine completed successfully, the debugger signal interface controller code block when executed causes the computer to perform a predetermined test action in order to cause a breakpoint to be set by the debugger code which is triggered when the processing unit performs said update of the status data. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 42:

Rajsuman discloses A computer program product as claimed in claim 41, wherein the debugger code is operable to issue a callback event to the debugger signal interface controller code block upon triggering of the breakpoint. **(Column 12, Lines 15-35)**

Regarding Claim 43:

Rajsuman discloses A computer program product as claimed in claim 35, wherein said stimulus and/or response signals are transferred between the debugger signal interface controller code block and the debugger code using Application Programming Interface (API) calls. **(Column 12, Lines 8-14)**

Regarding Claim 44:

Rajsuman discloses A computer program product as claimed in claim 36, wherein at least one of the software routines is written into the memory via the debugger code under the control of the computer executing the debugger signal interface controller code block. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 45:

Rajsuman discloses A computer program product as claimed in claim 35, wherein multiple processing units are provided, and a corresponding multiple of debugger signal interface controller code blocks are provided, each debugger signal interface controller code block when executed causes the computer to communicate with the same debugger code to cause their respective test actions to be performed. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 46:

Rajsuman discloses A computer program product as claimed in claim 35, wherein the timing of the execution of said software routines is co-ordinated with the sequence of verification tests. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 47:

Rajsuman discloses A computer program product as claimed in claim 35, wherein the system under verification comprises a plurality of components, each signal interface controller code block being associated with one of said components. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 48:

Rajsuman discloses A computer program product as claimed in claim 47, wherein the processing unit forms one of the components of the system under verification. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 49:

Rajsuman discloses A computer program product as claimed in claim 35, wherein the processing unit is provided externally to the system under verification. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 50:

Rajsuman discloses A computer program product as claimed in claim 35, wherein the processing unit comprises a representation of a processor on which the software routines are intended to be executed. **(Column 5, Lines 41-48. Figure 5)**

Regarding Claim 51:

Rajsuman discloses A computer program product as claimed in claim 35, wherein the system under verification comprises hardware simulator code when executed causes the computer in response to said one or more stimulus signals to generate said one or more response signals simulating a response of a data processing apparatus to said one or more stimulus signals if applied to said data processing apparatus. **(Column 5, Lines 41-48. Figure 5)**

(10) Response to Argument

Response to Argument – 101 Rejection

(10.1) Appellant argues that the claims recite statutory subject matter.

Examiners Answer:

Regarding Claims 1-51, the Examiner asserts that the current state of the claim language is such that a reasonable interpretation of the claims would not result in any useful, concrete or tangible result. The Examiner asserts that the claims do not indicate if the methods or apparatus are tangible methods or apparatus utilizing hardware, instead of an arrangement of software lacking tangible embodiment. As per Appellants specification the elements of the claims can be implemented in software as well as hardware. For example the processing unit discussed in the specification can be implemented in either hardware or software and as such the claims result in software per se which is not patentable. Appellants have stated that the processing unit utilizes physical signals however no citation of the specification has been provided to determine if the processing unit **can only be implemented in hardware**. Appellant's arguments regarding signals also reinforce the 101 rejections since signals are non-statutory.

Response to Argument – Prior Art Rejection

(10.2) Appellant argues that the reference does not disclose a debugger to control operation of a processing unit included in the hardware being tested which is also executing software routines as part of a software verification test.

Examiners Answer:

Column 5, Lines 4-7 of the reference state:

It is a further object of the present invention to provide a method and apparatus which allows a user to debug a fault in the cores of the SoC much more easily than the present day systems.

Column 11, Lines 53-63 of the reference state:

The vectors in the testbench data exercise various timing critical paths in SoC connecting different parts of SoC. As described, the design validation station of the present invention has all components of SoC, it is expected that the simulation testbench for timing verification will execute and produce results identical to the simulation. Any deviation from the simulation results identifies an error, that is debugged easily in the event based environment of this invention that is commensurate to design simulation environment.

Claim 5 of the reference states:

5. A method for design validation as defined in claim 4, wherein the simulation testbench of the individual cores has an event based data format, thereby facilitating a procedure to generate the test pattern signals for debugging of a fault in the cores of the SoC by the verification unit.

These sections indicate that the reference generates test pattern signals for debugging a fault which reads on the limitation as recited.

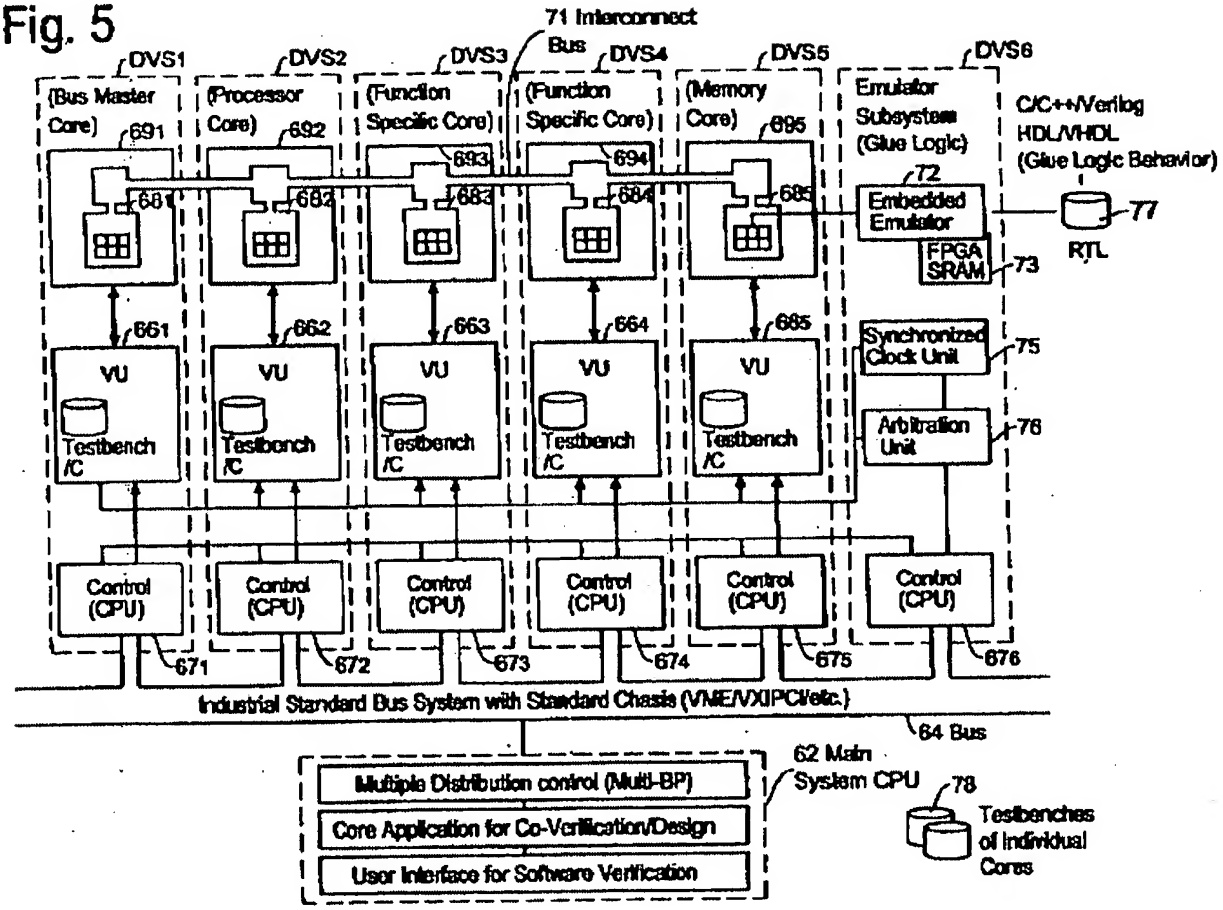
Furthermore, Column 9, Lines 16-24 states:

As the data is design simulation data, a defect free core performs exactly as predicted by the simulation. This response is observed and compared by the control CPU **67** in the verification unit **66**. Any deviation from the simulation is identified by the control CPU **67**. This allows to know the presence of any fault in the core IC on any verification unit (VU) **66**. This step allows to have a fault-free silicon IC of the core on a verification unit (VU) **66** before the SoC level design validation.

Appellants have further argued that the references discussion of a debugger does not control operation of a processing unit. However, this section indicates that the control CPU is configured to detect deviations, which further indicate a fault. Therefore, the reference states a control CPU detecting deviations which are considered faults and thereby facilitating the creation of test pattern signals to be generated to debug the cores. Consequently, the control CPU, for example, controls operation in at least detecting and debugging faults.

Appellants have argued on page 12 of the Appeal Brief that the claimed debugger synchronizes the hardware and software activities to achieve the co verification. In response to Appellants argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., **synchronization of activities as argued**) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). However, for the sake of argument Figure 5 of the reference shows a synchronization unit, element 75 reproduced below:

Fig. 5



Appellants further argue that **Rajsuman discloses the claimed debugger functionality** as can be seen on page 13 of the Appeal Brief, first paragraph and then states that this teaching in the reference is directed to detection of faults rather than the coordinated execution of the software and hardware tests of the claimed invention. However Column 7, Lines 15-25 of the reference states that the purpose of the references is coverification of which debugging is a key aspect as discussed above.

> An example of basic structure in the design validation station **50** is shown in the schematic diagram of FIG. **4B** for software/hardware co-development/verification. The design validation station **50** includes a plurality of verification units (VU) 66_1–66_N which are reconfigurable depending on pins of devices to be tested. The verification units 66_1–66_N are allocated to silicon ICs 68_1–68_N which [carry] are physical semiconductor devices carrying the function and circuit structure of the corresponding cores A–N to be integrated in the SoC to be evaluated.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the Examiner in the Related Appeals and Interferences section of this Examiner's Answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Saif Alhija, Patent Examiner

KAMINI SHAH
SUPERVISORY PATENT EXAMINER

Conferees:

Kamini Shah, Supervisory Patent Examiner

EDDIE C. LEE
SUPERVISORY PATENT EXAMINER

Eddie Lee, Supervisory Patent Examiner